

Mecanismo de upgrade de protocolo

O protocolo HTTP/1.1 fornece um mecanismo especial que pode ser usado para fazer upgrade de uma conexão já estabelecida para um protocolo diferente, utilizando o campo de cabeçalho Upgrade.

Esse mecanismo é opcional; ele **não pode** ser usado para exigir uma mudança de protocolo. Implementações podem optar por **não aproveitar** o upgrade mesmo que suportem o novo protocolo e, na prática, esse mecanismo é usado principalmente para **iniciar uma conexão WebSocket**.

Note também que o HTTP/2 proíbe explicitamente o uso desse mecanismo; ele é específico do HTTP/1.1.

Fazendo upgrade de conexões HTTP/1.1

O campo de cabeçalho Upgrade é usado por clientes para **convidar o servidor a trocar** para um dos protocolos listados, em ordem decrescente de preferência.

Como Upgrade é um cabeçalho do tipo hop-by-hop, ele também precisa ser listado no campo de cabeçalho Connection. Isso significa que uma requisição típica contendo Upgrade se pareceria com:



```
vbnet
GET /index.html HTTP/1.1
Host: www.exemplo.com
Connection: upgrade
Upgrade: exemplo/1, foo/2
```

Outros cabeçalhos podem ser exigidos dependendo do protocolo requisitado; por exemplo, upgrades para WebSocket permitem cabeçalhos adicionais para configurar detalhes da conexão WebSocket bem como oferecer certo grau de segurança ao abrir a conexão. Veja [Upgrade para uma conexão WebSocket](#) para mais detalhes.

Se o servidor decidir fazer o upgrade da conexão, ele envia de volta um código de status **101 Switching Protocols**, com um cabeçalho Upgrade que especifica o(s) protocolo(s) sendo usados.

Se ele **não fizer (ou não puder fazer)** o upgrade da conexão, ele ignora o cabeçalho Upgrade e envia uma resposta regular (por exemplo, 200 OK).

Logo após enviar o status 101, o servidor pode começar a falar no novo protocolo, realizando quaisquer handshakes específicos adicionais, se necessário. Efetivamente, a conexão se torna um canal bidirecional assim que a resposta do upgrade estiver completa, e a requisição que iniciou o upgrade pode ser concluída sobre o novo protocolo.

Usos comuns para esse mecanismo

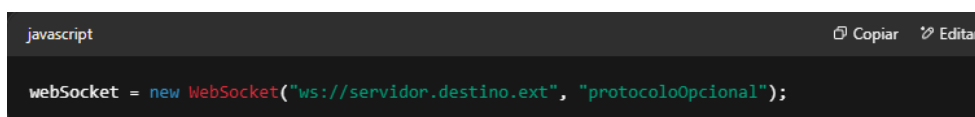
Aqui estão os casos de uso mais comuns para o cabeçalho Upgrade.

Fazendo upgrade para uma conexão WebSocket

De longe, o caso de uso mais comum para fazer upgrade de uma conexão HTTP é para utilizar **WebSockets**, que são sempre implementados por meio de upgrade de uma conexão HTTP ou HTTPS.

Tenha em mente que se você está abrindo uma nova conexão usando a API WebSocket, ou qualquer biblioteca que implemente WebSockets, a maior parte (ou tudo) disso é feito automaticamente para você.

Por exemplo, abrir uma conexão WebSocket é feito com um único método:



```
javascript
WebSocket = new WebSocket("ws://servidor.destino.ext", "protocoloOpcional");
```

O construtor `WebSocket()` faz todo o trabalho de criar a conexão HTTP/1.1 inicial e então lidar com o processo de handshake e upgrade para você.

Se você precisa criar uma conexão WebSocket do zero, terá de lidar com o processo de handshake manualmente. Após criar a sessão HTTP/1.1 inicial, você precisa solicitar o upgrade adicionando à requisição padrão os cabeçalhos Upgrade e Connection, como segue:

```
makefile Copiar Editar
Connection: Upgrade
Upgrade: websocket
```

Cabeçalhos específicos para WebSocket

Os seguintes cabeçalhos estão envolvidos no processo de upgrade para WebSocket. Além dos cabeçalhos Upgrade e Connection, os demais são geralmente opcionais ou tratados automaticamente pelo navegador e servidor durante a negociação.

Sec-WebSocket-Extensions

Especifica uma ou mais extensões de protocolo WebSocket a serem solicitadas ao servidor. É permitido usar mais de um cabeçalho Sec-WebSocket-Extensions em uma requisição; o efeito é o mesmo que incluir todas as extensões listadas em um único cabeçalho.

Exemplo:

```
makefile Copiar Editar
Sec-WebSocket-Extensions: superspeed, colormode; depth=16
```

Uma lista separada por vírgulas de extensões a serem requisitadas (ou às quais se está disposto a dar suporte). Essas extensões devem ser escolhidas do [registro de extensões WebSocket da IANA](#). Extensões que usam parâmetros devem separá-los com ponto e vírgula.

Sec-WebSocket-Key

Fornece informações ao servidor necessárias para confirmar que o cliente está autorizado a solicitar o upgrade para WebSocket.

Esse cabeçalho pode ser usado quando clientes inseguros (HTTP) desejam fazer upgrade, a fim de oferecer algum grau de proteção contra abusos.

O valor da chave é calculado usando um algoritmo definido na especificação WebSocket, então **isso não fornece segurança real**, mas ajuda a impedir que clientes que **não são WebSocket** solicitem inadvertidamente (ou por erro) uma conexão WebSocket.

Esse cabeçalho é adicionado automaticamente por clientes que optam por usá-lo; **não pode ser adicionado usando os métodos `fetch()` ou `XMLHttpRequest.setRequestHeader()`**.

O servidor incluirá, na resposta, um cabeçalho Sec-WebSocket-Accept, com um valor calculado com base na chave fornecida.

Sec-WebSocket-Protocol

O cabeçalho Sec-WebSocket-Protocol especifica um ou mais **subprotocolos WebSocket** que você deseja usar, em ordem de preferência.

O primeiro que for suportado pelo servidor será selecionado e retornado no mesmo cabeçalho na resposta.

Você pode usar esse cabeçalho mais de uma vez; o efeito é equivalente ao de usar uma lista de subprotocolos separados por vírgula em um único cabeçalho.

Exemplo:

```
makefile
Copiar Editar
Sec-WebSocket-Protocol: chat, superchat
```

Os subprotocolos podem ser escolhidos do [registro de subprotocolos WebSocket da IANA](#) ou podem ser nomes personalizados acordados entre cliente e servidor.

Sec-WebSocket-Version

CABEÇALHO DE REQUISIÇÃO

Especifica a **versão do protocolo WebSocket** que o cliente deseja usar, para que o servidor possa confirmar se ela é suportada.

A versão mais recente deve ser escolhida com base no [registro de versões WebSocket da IANA](#). A versão final mais recente do protocolo WebSocket é a **versão 13**.

CABEÇALHO DE RESPOSTA

Se o servidor **não puder se comunicar** usando a versão especificada, ele responderá com um erro (como 426 Upgrade Required) incluindo no cabeçalho Sec-WebSocket-Version uma lista separada por vírgulas das versões que ele suporta.

Se ele suportar a versão requisitada, **esse cabeçalho não será incluído** na resposta.

Cabeçalhos somente de resposta

A resposta do servidor pode incluir os seguintes:

Sec-WebSocket-Accept

Incluído na mensagem de resposta do servidor durante o processo de handshake de abertura, quando o servidor estiver disposto a iniciar uma conexão WebSocket.

Aparecerá **no máximo uma vez** nos cabeçalhos da resposta.

Se um cabeçalho Sec-WebSocket-Key foi fornecido, o valor deste cabeçalho é calculado **concatenando a chave fornecida** com a string:

```
arduino
Copiar Editar
"258EAF5 - E914 - 47DA - 95CA - C5AB0DC85B11"
```

Depois, é calculado o hash SHA-1 dessa string concatenada, gerando um valor de 20 bytes. Esse valor é então codificado em Base64 para obter o valor final do Sec-WebSocket-Accept.